

Desarrollo de una aplicación informática para el análisis de imágenes térmicas



Autor: Alejandro Sepulcre Vaz

Director: Gonzalo Pajares Martisanz

Curso 2017-2018

Trabajo de fin de grado del Grado en Ingeniería del Software
Facultad de Informática, Universidad Complutense de Madrid

Índice

<i>Listado de abreviaturas</i>	2
<i>Listado de figuras</i>	3
<i>Resumen</i>	4
<i>Abstract</i>	5
1.- Introducción	6
1.1 <i>Consideraciones preliminares</i>	6
1.2. <i>Objetivos</i>	7
1.3. <i>Plan de trabajo</i>	8
1.4. <i>Estructura de la memoria</i>	8
1.5. <i>Preliminary considerations</i>	8
1.6. <i>Objectives</i>	10
1.7. <i>Work plan</i>	10
2.- Funcionalidades de la aplicación	11
2.1 <i>Preprocesamiento de imágenes</i>	11
2.2 <i>Método de Lucas-Kanade: flujo óptico</i>	12
2.3 <i>Detección de objetos en movimiento</i>	14
3.- Diseño e implementación	19
3.1 <i>Diseño</i>	19
3.2 <i>Implementación</i>	21
4.- Resultados	22
5.- Conclusiones y trabajo futuro	32
6.- Conclusions and future work	33
Bibliografía	36
Anexo I: Código base Lucas-Kanade	37
Anexo II: Manual de usuario	39

Listado de abreviaturas

HSV *Hue, Saturation, Value*

RGB *Red, Green, Blue*

GUI *Graphic User Interface*

GUIDE *Graphic User Interface Development Environment*

Listado de figuras

Figura 1. Imágenes consecutivas de vídeo: (a) referencia; (b) objetos en movimiento	11
Figura 2. Representación de una secuencia temporal (t) de imágenes con dimensiones espaciales x e y.	12
Figura 3. Representación del flujo óptico para las imágenes mostradas en la figura 1.	14
Figura 4. Histogramas de intensidad con varios umbrales.	15
Figura 5. Imagen binarizada a partir de la imagen de la figura 1 con $T = 220$.	17
Figura 6. Imagen sin ruido.	18
Figura 7. Identificación de objetos	19
Figura 8. Etiquetado o coloración de objetos	19
Figura 9. GUIDE de Matlab	20
Figura 10. Diagrama de clase	20
Figura 11. Caso 1: imágenes de una secuencia de vídeo (personas en movimiento).	22
Figura 12. Diferencias entre dos imágenes mediante Lucas-Kanade.	23
Figura 13. (a) Imagen binarizada a partir de la imagen de la figura 12; (b) la misma imagen sin ruido; (c) imagen con los objetos etiquetados e identificados por color.	23
Figura 14. Flujo óptico para las imágenes del caso 1.	24
Figura 15. Caso 2: imágenes de una secuencia de vídeo (personas y vehículo en movimiento).	25
Figura 16. Diferencias entre dos imágenes mediante Lucas-Kanade.	25
Figura 17. (a) Imagen binarizada a partir de la imagen de la figura 16; (b) la misma imagen sin ruido; (c) imagen con los objetos etiquetados e identificados por color.	26
Figura 18. Flujo óptico para las imágenes del caso 2.	26
Figura 19. Caso 3: imágenes de una secuencia de vídeo (personas y vehículo en movimiento con el grupo de personas superado).	27
Figura 20. Diferencias entre dos imágenes mediante Lucas-Kanade.	28
Figura 21. (a) Imagen binarizada a partir de la imagen de la figura 20; (b) imagen con los objetos etiquetados e identificados por color.	28
Figura 22. Flujo óptico para las imágenes del caso 3.	29
Figura 23. Caso 4: escena con todos los elementos estáticos.	30
Figura 24. Lucas-Kanade Caso 4	30
Figura 25. (a) Imagen binarizada a partir de la imagen de la figura 24; (b) imagen binaria con eliminación de ruido.	31
Figura 26. Flujo óptico para las imágenes del caso 4.	32

Resumen

En el campo de la percepción computacional y más concretamente en el ámbito de la Visión por Computador, uno de los principales campos es el análisis del movimiento en la escena, bien por parte de los objetos en ella, por el movimiento de la cámara o por ambos a la vez. En el presente trabajo se desarrolla una aplicación informática con el objeto de detectar el movimiento entre dos imágenes consecutivas, extraídas de un vídeo captado con una cámara termográfica, para su posterior uso en tareas tales como videovigilancia o monitorización de objetos basadas en diferencias térmicas de los elementos en movimiento respecto de la escena.

Con este fin, se han aplicado métodos propios de procesamiento de imágenes con técnicas de programación basada en los principios de la Inteligencia Artificial para crear una aplicación que permita obtener resultados para los objetivos planteados.

El desarrollo de la aplicación se realiza en lenguaje de programación Matlab, para procesar dos imágenes consecutivas de una secuencia utilizando el método de Lucas-Kanade, junto con técnicas multi-umbral, lo que permite la detección del movimiento, junto con la estima de la magnitud y sentido de éste, en lo que técnicamente se conoce como flujo óptico. Más específicamente, la aplicación consta de dos fases, de forma que en la primera se detecta el movimiento y su intensidad mediante Lucas-Kanade y en la segunda los objetos que aparecen en la zona de movimiento mediante las técnicas de umbralización.

Palabras Clave: Método Lucas-Kanade, Multi-umbral, Flujo óptico, Visión por Computador, Matlab, procesamiento imágenes.

Abstract

In the field of computational perception and more specifically in the field of Computer Vision, one of the main fields is the analysis of movement in the scene, either by the objects in it, by the movement of the camera or by both at the same time. In the present work a computer application is developed in order to detect the movement between two consecutive images, extracted from a video captured with a thermal camera, for its later use in tasks such as video surveillance or monitoring of objects based on thermal differences of the elements in movement with respect to the scene.

To this end, methods of image processing with programming techniques based on the principles of Artificial Intelligence have been applied in order to create an application that allows obtaining results for the proposed objectives.

The development of the application is done in Matlab programming language, to process two consecutive images of a sequence using the Lucas-Kanade method, together with multi-threshold techniques, which allows the detection of movement, together with the estimation of the magnitude and sense of it, in what is technically known as optical flow. More specifically, the application consists of two phases, so that in the first movement and intensity is detected by Lucas-Kanade and in the second the objects that appear in the movement zone by means of thresholding techniques.

Keywords: Lucas-Kanade method, Multi-thresholding, Optical flow, Computer Vision, Matlab, image processing

1.- Introducción

1.1. Consideraciones preliminares

En el campo del tratamiento de imágenes y más concretamente dentro de la Visión por Computador, un campo importante de aplicación es justamente el análisis del movimiento de los objetos en la escena, bien por el propio movimiento de éstos, bien por el movimiento de la cámara que captura las imágenes o por ambas cosas a la vez. En el presente trabajo se plantea detectar el movimiento entre dos imágenes consecutivas, obtenidas a partir de un vídeo termográfico para su posterior uso en diversas aplicaciones, tales como vídeo-vigilancia o monitorización de esos objetos.

Con este fin, se han aplicado técnicas específicas de procesamiento de imágenes, en el contexto de la Inteligencia Artificial, y programación para crear una aplicación que permita obtener resultados para los objetivos planteados.

Para ello se ha desarrollado una aplicación en lenguaje Matlab para procesar dichas imágenes utilizando el método de Lucas-Kanade (1981) y varios métodos de multi-umbralización, lo que permite identificar si existe movimiento en la escena en base al contenido de las imágenes, además de así poder estimar el flujo óptico (Pajares y Cruz, 2007). Una vez detectado el movimiento, en lo que constituye una primera fase de la aplicación desarrollada, el objetivo siguiente consiste en la identificación de los objetos presentes en la zona del movimiento detectado mediante técnicas de segmentación mediante umbral.

En el campo del análisis del movimiento y desde un punto de vista práctico existen tres grandes grupos de problemas:

1. *Detección del movimiento*, cuya finalidad es determinar cualquier movimiento desplazamiento de los objetos en la escena tomando como base una imagen de referencia y otra tomada en un instante posterior de la misma escena. Es útil en el campo de la seguridad y monitorización. En este caso se suele utilizar una simple cámara estática.
2. *Detección y localización de los objetos en movimiento*, una cámara se sitúa en una posición estática y los objetos se mueven en la escena, o la cámara se mueve y los

objetos son estáticos o ambas cosas a la vez. El problema característico consiste en la detección del objeto, la detección de la trayectoria de su movimiento y la predicción de su futura trayectoria. Algunos ejemplos de interés en este campo son: la evolución de una masa nubosa mediante secuencias de imágenes por satélite incluyendo la predicción de su movimiento; también el control y predicción del tráfico de vehículos en una ciudad, etc.

3. *Obtención de propiedades 3D* de los objetos a partir de un conjunto de proyecciones 2D adquiridas en distintos instantes de tiempo del movimiento de los objetos y con diferentes posicionamientos o puntos de vista de las cámaras en relación a la escena.

En este trabajo se aborda exactamente el problema de la detección del movimiento en imágenes térmicas, además de la estimación del flujo óptico. Para ello, en la zona donde se detecta el movimiento se aplican técnicas de umbralización para determinar los objetos que han causado el movimiento en dicha zona. El objetivo final resultante de la combinación de ambas fases es su posterior uso en diversas aplicaciones, como se ha indicado previamente.

1.2. Objetivos

Tomando como referencia los planteamientos anteriores, se definen los siguientes objetivos:

1. Desarrollar una interfaz gráfica simple e intuitiva, utilizando patrones de diseño según los planteamientos de la Ingeniería del Software.
2. Analizar y procesar imágenes de vídeo tomadas por una cámara termográfica.
3. Implementar el método de Lucas-Kanade para detección del movimiento y estimación del flujo óptico a partir de dos imágenes consecutivas de una secuencia de vídeo.
4. Identificar y valorar el comportamiento de diferentes métodos de multi-umbralización para determinar los más apropiados desde el punto de vista de la detección de los objetos que han originado el movimiento en las zonas donde el flujo óptico presenta una magnitud relevante.
5. Implementar una aplicación utilizando la herramienta de desarrollo Matlab y su toolbox Image Processing.

1.3. Plan de trabajo

Para cumplir con los objetivos expuestos, se propone el siguiente plan de trabajo, desglosado por objetivos y duración de los mismos teniendo en cuenta la duración aproximada de ocho meses del proyecto, que hace un total de 32 semanas

Objetivo	Duración(Semanas)
Desarrollo de la interfaz de usuario	2
Pre-procesamiento del vídeo y funcionalidades complementarias	5
Implementación del método Lucas-Kanade	4
Cálculo del flujo óptico	7
Diversas pruebas con los diferentes métodos de multi-umbralización	4
Redacción de la memoria	10

1.4. Estructura de la memoria

La memoria se estructura según se describe a continuación. En la sección 1 se desarrolla la Introducción del trabajo. La sección 2 describe las funcionalidades de la aplicación, incluyendo la base conceptual de la misma. Se incluye la descripción del método de Lucas-Kanade para la determinación del flujo óptico además de los métodos basados en multi-umbralización para identificar los objetos que han originado el movimiento detectado. En la sección 3 se describe el diseño e implementación básica sobre la que se sustenta la aplicación. En la sección 4 se analizan algunos resultados significativos obtenidos en base a los métodos desarrollados. En la sección 5 se proporcionan las conclusiones más relevantes, así como algunas indicaciones para posibles mejoras y trabajos futuros.

Se incluyen sendos Anexos I y II. En el Anexo I se describe brevemente la base de la implementación del algoritmo de Lucas-Kanade y su código correspondiente. Finalmente, en el Anexo II se incluye el manual de usuario para manejo de la aplicación desarrollada bajo la perspectiva de su encapsulación en el interfaz de usuario también desarrollado y que forma parte igualmente de la aplicación.

1.5 Preliminary considerations

In the field of image processing and more specifically within Computer Vision, an important field of application is precisely the analysis of the movement of objects in the scene, either by the movement of these, or by the movement of the camera that captures the

images or both simultaneously. In the present work the movement is detected between two consecutive images, obtained from a thermographic video camera for its later use in diverse applications, such as video surveillance or monitoring of those objects.

To this end, specific techniques of image processing have been applied, in the context of Computer Vision inside Artificial Intelligence. They have been programed to create an application that allows obtaining results for the proposed objectives.

Thus, an application in Matlab language has been developed to process these images using the Lucas-Kanade (1981) approach and several multi-thresholding methods, which allows to identify if there is movement in the scene based on the content of the images, in addition to being able to estimate the optical flow (Pajares and Cruz, 2007). Once the movement has been detected, in what constitutes a first phase of the developed application, the following objective consists of the identification of objects present in the detected movement area by means of thresholding segmentation techniques.

In the field of movement analysis and from a practical point of view there are three main groups of problems:

1. *Detection of movement*, whose purpose is to determine any movement of objects in the scene, based on a reference image and another captured later from the same scene. It is useful in the field of security and monitoring. In this case, a simple static camera is usually used.
2. *Detection and location of objects in motion*, a camera is placed in a static position and objects move in the scene, or the camera moves, and the objects are static or both at the same time. The main problem consists in the detection of objects, together with their trajectories of movements and the prediction of their future trajectories. Some examples of interest in this field are: the evolution of a cloud mass through sequences of satellite images including the prediction of its movement; also, the control and prediction traffic in a city among others. They fall on the topic of tracking.
3. *Obtaining 3D properties* of the objects from a set of 2D projections acquired at different time instants of the movement of the objects and with different positioning or points of view of the cameras in relation to the 3D scene.

The problem of motion detection in thermal images is the focus of this work, where optical flow computation becomes a relevant issue. In addition, in the area where motion is detected, thresholding techniques are applied to determine the objects that caused the movement in the area. The final goal resulting from the combination of both phases is its subsequent use in various applications, as previously indicated.

1.6 Objectives

Considering the previous approaches as a reference, the following objectives are defined in order to:

1. Develop a simple and intuitive user graphic interface, using design patterns according to the approaches provided by the software engineering discipline.
2. Analyse and process video images captured by a thermal imager in the form of video images.
3. Implement the Lucas-Kanade approach, designed for motion detection and optical flow estimation from two consecutive images of a video sequence.
4. Identify and evaluate the behaviour of different multi-thresholding methods to determine the most appropriate threshold, from the point of view of object detection, causing the movement in the areas where the optical flow has a relevant magnitude.
5. Implement an application using the Matlab development tool and its Image Processing toolbox.

1.7 Work plan

To fulfil the objectives above, the following work plan is proposed, broken down by objectives and duration of them considering the approximate duration of eight months of the project, which makes a total of 32 weeks.

Objective	Duration (weeks)
Development of the user interface	2
Video pre-processing and complementary functionalities	5
Implementation of the Lucas-Kanade method	4
Optical flow computation	7
Various tests with the different multi-thresholding methods	4
Report generation	10

2.- Funcionalidades de la aplicación

A continuación se describen las funcionalidades que conforman la aplicación desarrollada, a saber: a) Pre-procesamiento de imágenes; b) método de detección del flujo óptico mediante el algoritmo de Lucas-Kanade y c) detección de objetos en movimiento mediante técnicas de umbralización.

2.1 Pre-procesamiento de imágenes

La base para la determinación del movimiento y posterior cómputo del flujo óptico, requiere disponer de dos imágenes, obtenidas a partir de un vídeo, ambas capturadas con una cierta diferencia temporal. Una de las imágenes constituye lo que se conoce como imagen de referencia y la otra imagen se denomina imagen de diferencias con respecto a la de referencia. En el caso que nos ocupa las imágenes proceden de una secuencia de vídeo capturadas con una cámara térmica Flir-One (2018). La Figura 1 muestra en (a) una imagen de referencia dentro de una secuencia de vídeo y en (b) una segunda imagen para comparación con la de referencia conteniendo los objetos de la escena.

Como bien es sabido, las imágenes térmicas reflejan diferencias térmicas entre los objetos en la escena sobre la base de que todos los objetos por el mero hecho de poseer una temperatura por encima del cero absoluto emite radiación en la banda espectral correspondiente al térmico, aproximadamente dentro del rango de longitudes de onda correspondiente a $8 - 14\mu\text{m}$.



Figura 1. Imágenes consecutivas de vídeo: (a) referencia; (b) objetos en movimiento.

Tras la carga de las imágenes deseadas éstas se transforman a una imagen en escala de grises mediante la transformación del modelo de color RGB, que es el original de las imágenes utilizadas, al modelo *HSI* (Pajares y Cruz, 2007). De esta forma, se consigue un

aumento del contraste de la imagen de intensidad (I) lo que facilita posteriormente la segmentación de la imagen tanto desde el punto de vista de la detección del movimiento como desde la aplicación de los diferentes métodos de multi-umbralización.

2.2 Método de Lucas-Kanade: flujo óptico

El método de Lucas-Kanade (1981) desarrolla una técnica diferencial ampliamente usada para la detección del movimiento y estimación del flujo óptico, el principal concepto es la suposición de que el flujo es esencialmente constante en las proximidades espaciales a un píxel en la imagen bajo consideración. Bajo este planteamiento, la base del mismo consiste en determinar tanto variaciones espaciales en las direcciones x e y de la escena como variaciones temporales. Se introduce así una tercera dimensión en el tratamiento de las imágenes, cual es la temporal. La imagen de la figura 2, muestra una representación de una secuencia de imágenes en el tiempo (vídeo).

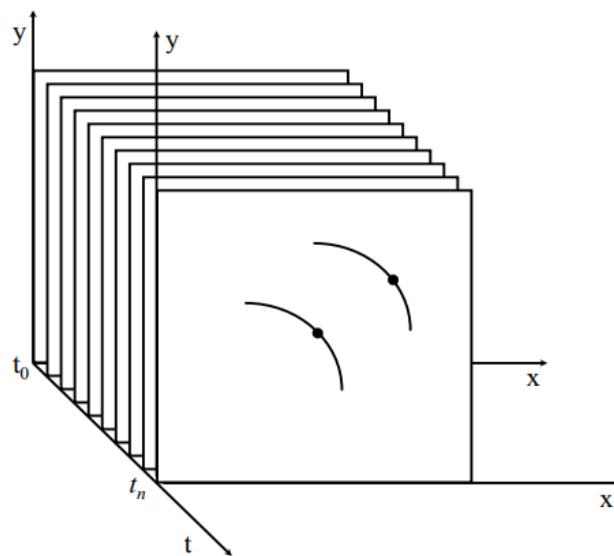


Figura 2. Representación de una secuencia temporal (t) de imágenes con dimensiones espaciales x e y .

La solución se plantea a nivel de píxel analizando las proximidades de cada píxel alrededor de una ventana, se plantean las correspondientes restricciones, que derivan en la formulación de lo que se conoce como ecuación del flujo óptico (Pajares y Cruz, 2007), que se resuelve por mínimos cuadrados, como se verá más adelante.

En consecuencia, el método de Lucas-Kanade asume que el desplazamiento de los contenidos de una imagen entre dos instantes próximos en el tiempo (*frames*) es pequeño y aproximadamente constante en un conjunto cercano al punto p (píxel) en consideración.

Con lo que se puede asumir que la ecuación del flujo óptico, ecuación (1), se cumple para una ventana de vecindad centrada en el píxel p .

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \mathbf{G} = 0 \quad (1)$$

Bajo la suposición de desplazamientos espaciales pequeños, la ecuación anterior se puede desarrollar mediante desarrollo de Taylor, resultando en la ecuación (2) que se muestra a continuación.

$$-f_t = f_x \frac{dx}{dt} + f_y \frac{dy}{dt} \quad (2)$$

Planteada la ecuación anterior, el siguiente problema es obtener la velocidad $\mathbf{v} = \left(\frac{dx}{dt}, \frac{dy}{dt} \right)$ asumiendo que las derivadas espaciales f_x y f_y así como la temporal f_t en relación a la imagen de intensidad, ahora representada por f , son todas cantidades medibles o al menos puede obtenerse una medida aproximada de ellas.

En la ecuación (2) se asume que durante el movimiento, la intensidad de la imagen permanece constante. Teniendo en cuenta esta suposición y considerando el planteamiento de Lukas-Kanade, la ecuación (2) se expresa como sigue,

$$\begin{bmatrix} \partial^2 f / \partial x^2 & \partial^2 f / \partial x \partial y \\ \partial^2 f / \partial x \partial y & \partial^2 f / \partial y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \frac{\partial \mathbf{G}}{\partial t} \quad (3)$$

Considerando, un entorno de vecindad Ω alrededor de cada píxel sobre el que se extiende el sumatorio, la ecuación (3) se puede expresar como sigue,

$$\begin{bmatrix} \sum_{\Omega} f_x^2 & \sum_{\Omega} f_x f_y \\ \sum_{\Omega} f_x f_y & \sum_{\Omega} f_y^2 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} \sum_{\Omega} f_t f_x \\ \sum_{\Omega} f_t f_y \end{bmatrix} \equiv \mathbf{A} \mathbf{v} = \mathbf{b} \quad (4)$$

La solución de la ecuación (4) se obtiene por mínimos cuadrados (Pajares y Cruz, 2007), obteniéndose finalmente la solución dada por la ecuación (5).

$$\mathbf{v} = \left(A^T A \right)^{-1} A^T b \quad (5)$$

Donde A^T es la matriz transpuesta de A y \mathbf{v} es la solución del problema, que a la vez es el propio flujo óptico.

En la figura 3 se muestra un ejemplo ilustrativo relativo a la detección del flujo óptico, mediante la aplicación de Lucas-Kanade a partir del par de imágenes representativas mostradas en la figura 1. Obsérvese en este ejemplo la intensidad de las flechas amarillas que son las que representan los valores correspondientes al flujo óptico \mathbf{v} , de forma que a mayor intensidad de color mayor magnitud del flujo. Además, la dirección de las mismas expresan también la dirección del movimiento.

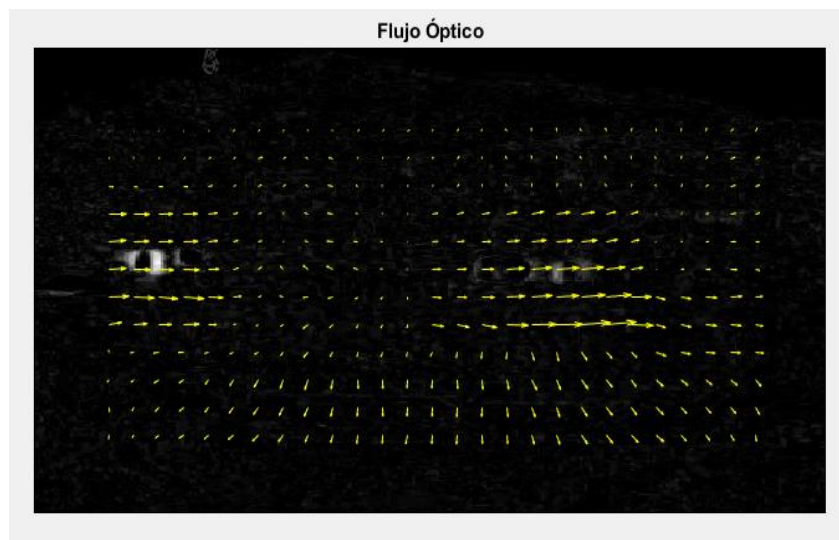


Figura 3. Representación del flujo óptico para las imágenes mostradas en la figura 1.

2.3 Detección de objetos en movimiento

Para la identificación de los objetos en movimiento en la imagen se procede a tratar la imagen como se describe a continuación. Así, mediante Lucas-Kanade, como se ha indicado previamente, se obtienen las zonas donde se ha detectado una intensidad de movimiento relevante, que se determina por la magnitud del flujo óptico. Pues bien, mediante el proceso de segmentación que se describe a continuación se extraen objetos relevantes en la escena, que además se etiquetan con el fin de asociar los mismos con las zonas donde se ha detectado el movimiento. De esta forma se asocia movimiento con objetos que supuestamente lo han provocado.

El proceso de segmentación para la identificación de objetos en las zonas de elevada intensidad de flujo óptico consta de tres procesos fundamentales, a saber: a) binarización; b) eliminación de ruido y c) etiquetado de componentes conexas.

a) *Binarización de la imagen*

La binarización de imágenes consiste en segmentar una imagen de diferentes intensidades o grises, separando los píxeles que se sitúan por debajo de un determinado valor de intensidad, que se denomina umbral frente a los que se sitúan por encima. Este proceso se realiza sobre el histograma de la imagen, que consiste en obtener la distribución de los niveles de intensidad según la frecuencia de aparición de éstos en la imagen. A partir del histograma se determina un nivel de intensidad que separa el histograma en dos partes, tal es el caso mostrado en la figura 4(a) con T . En este caso, la imagen resultante aparecería con dos regiones identificadas como blanco y negro, de aquí el término de binarización. Si se realiza una separación de acuerdo a dos niveles de intensidad (T_1 , T_2), figura 4(b), el resultado es una imagen donde se distinguen tres regiones diferentes según las tres regiones que establecen estos dos valores de intensidad. Los valores mencionados (T , T_1 , T_2) se denominan umbrales y en el caso en el que haya más de uno como es el caso de la imagen de la figura (b), el proceso de segmentación se denomina multi-umbralización, que es el utilizado en este trabajo, si bien se ha reducido a la utilización de un único umbral por razones operativas.

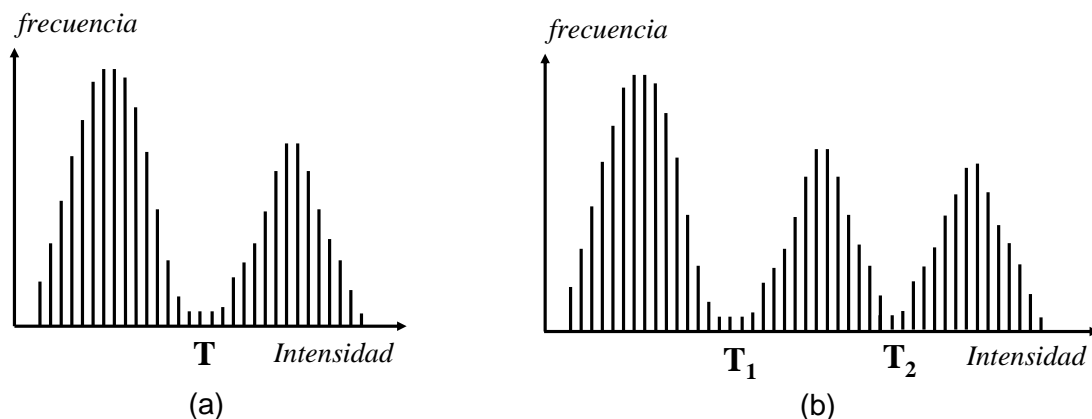


Figura 4. Histogramas de intensidad con varios umbrales.

Uno de los problemas que se plantean es determinar los mejores valores de umbral para que la imagen resulte efectiva. Con tal propósito, Otsu (1979) propuso un método para obtener un umbral óptimo de forma automática maximizando la varianza entre las clases (lóbulos) del histograma, figura 4, mediante una búsqueda exhaustiva de distribuciones de niveles sobre el propio histograma de la imagen.

Como bien es sabido, una imagen es una función bidimensional de la intensidad del nivel de gris o intensidad, y contiene N píxeles cuyos niveles de gris se encuentran entre 0 y L . El número de píxeles con nivel de gris i se denota como f_i , y la probabilidad de ocurrencia del nivel de gris i en la imagen está dada por la ecuación (6).

$$P(i) = \frac{f_i}{N} \quad (6)$$

Considerando un histograma bimodal, figura 4(a), se supone que la imagen posee un histograma formado por la suma de dos densidades de probabilidad gaussianas donde cada gaussiana se aproxima a uno de los lóbulos. Esto hace suponer que a medida que las gaussianas se asemejan al histograma real, las desviaciones estándar deben disminuir y como consecuencia de ello se debe elegir aquél umbral que minimice la suma de las varianzas de los dos lóbulos del histograma (Otsu, 1979).

Dada una imagen con L niveles de intensidad y asumiendo que el umbral buscado es T para el caso de una imagen bimodal, las probabilidades acumuladas hasta T y desde T hasta L resultan ser,

$$w_1(t) = \sum_{z=1}^T P(z) \text{ y } w_2(t) = \sum_{z=T+1}^L P(z) \quad (7)$$

Las medias y varianzas asociadas se obtienen como sigue,

$$\mu_1(t) = \sum_{z=1}^T zP(z) \text{ y } \mu_2(t) = \sum_{z=T+1}^L zP(z) \quad (8)$$

$$\sigma_1^2(t) = \sum_{z=1}^T (z - \mu_1(t))^2 \frac{P(z)}{w_1(t)} \text{ y } \sigma_2^2(t) = \sum_{z=T+1}^L (z - \mu_2(t))^2 \frac{P(z)}{w_2(t)} \quad (9)$$

Finalmente se obtiene la varianza ponderada,

$$\sigma_w^2(t) = w_1(t)\sigma_1^2(t) + w_2(t)\sigma_2^2(t) \quad (10)$$

Se elige el umbral T que corresponde al nivel de intensidad que proporcione la mínima varianza ponderada. La aplicación del método de Otsu a la imagen de la figura 1, permite obtener un valor de umbral $T = 220$, con el que se segmenta la imagen original para generar la imagen binarizada mostrada en la imagen de la figura 5, donde aparecen los objetos en movimiento detectados mediante el método de Lucas-Kanade y cuyo valor de umbral se sitúa por encima del valor T indicado previamente. La aplicación del método multi-umbral permite obtener diferentes valores, si bien en el caso que nos ocupa no resulta

operativo dada la naturaleza de la imagen original y la falta de contraste suficiente, que permita separar los lóbulos del histograma con suficientes garantías. En cualquier caso, el proceso multi-umbral es un caso mono-umbral en el que los lóbulos separados por un único umbral se vuelven a separar aplicando el método de Otsu para cada lóbulo separado. En el presente trabajo se utiliza el procedimiento multi-umbral, si bien particularizado para un único umbral, que en definitiva se reduce a la detección de umbral único. La figura 5 muestra una imagen binaria como resultado de aplicar el método de Otsu sobre la imagen de la figura 1, para la que se obtiene un valor de umbral $T = 220$.

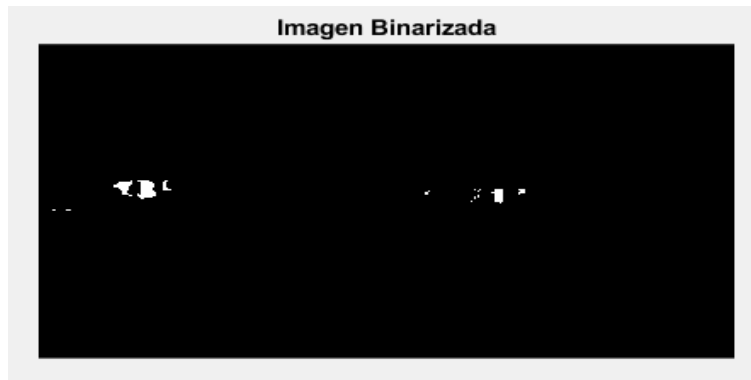


Figura 5. Imagen binarizada a partir de la imagen de la figura 1 con $T = 220$.

b) Eliminación de ruido

Una vez obtenida la imagen binaria es conveniente realizar una operación de eliminación de píxeles espurios sobre la misma y por tanto eliminar píxeles que sean innecesarios, que en definitiva proceden de la existencia de niveles de intensidad no deseados y de ruido. Con tal propósito se aplican técnicas basadas en morfología matemática (Pajares y Cruz, 2007), de forma que se eliminen píxeles aislados y componentes conexas relativamente pequeñas, formadas por un número inferior a P píxeles, siendo P un límite preestablecido, que en nuestro caso se ha fijado en 50 píxeles. Además se aplica la operación morfológica de apertura, que rellena pequeños huecos en las regiones conexas, así como una limpieza en los bordes de éstas.

El resultado es una imagen cuyo contenido se centra exclusivamente en las regiones de interés donde el movimiento resulta significativo. En la figura 6 se muestra el resultado de aplicar estas operaciones sobre la imagen de la figura 5 donde se puede observar fácilmente cómo los objetos de interés han quedado aislados, mientras determinados píxeles espurios han sido eliminados, lo que facilita posteriormente el etiquetado de las regiones que han quedado finalmente.

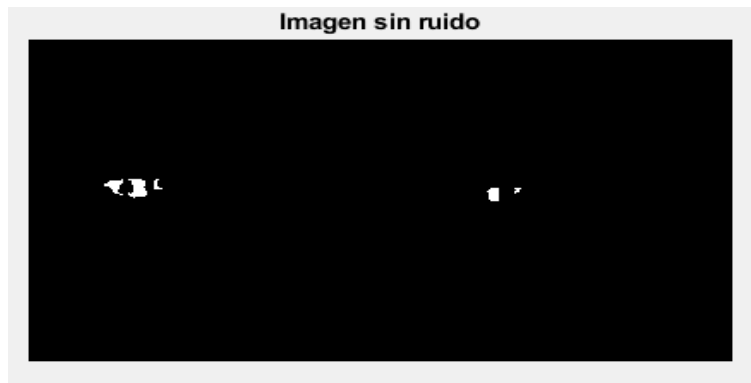


Figura 6. Imagen sin ruido.

c) Etiquetado de componentes conexas

Se parte del hecho de que la imagen está binarizada (imagen de ceros y unos). El objetivo consiste en etiquetar las componentes conexas de forma que el resultado final sea una región. El concepto de componentes conexas es el siguiente: todos los píxeles que tienen valor binario “1” y están conectados entre sí por un camino o conjunto de píxeles todos con el valor binario “1” (en el caso de imágenes se corresponde con el valor de 255) se les asigna la misma etiqueta identificativa, que debe ser única en cada región a la cual pertenecen los píxeles y constituye su identificador.

Un algoritmo clásico de etiquetado de componentes conexas es el desarrollado por Haralick y Shapiro (1992), cuyo fundamento estriba en el tipo de conectividad, que básicamente es de dos tipos, conectividad-4 y conectividad-8. El algoritmo procesa una fila de la imagen en cada instante y asigna nuevas etiquetas al primer píxel de cada componente propagando la etiqueta de un píxel a sus vecinos a la derecha o debajo de él. En definitiva, el etiquetado de componentes conexas es una aplicación algorítmica de la teoría de grafos donde subconjuntos de componentes conexas reciben etiquetas únicas según una heurística dada, en este caso la heurística se fundamenta en la igualdad de los valores blancos (marcados con 1) en la imagen binaria. De esta forma se van uniendo píxeles contiguos con los mismos niveles de intensidad y conectados según la región de vecindad definida.

La figura 7 muestra los resultados sobre identificación de objetos diferenciados por diferentes niveles de intensidad, que en realidad se corresponden con diferentes etiquetas asignadas por la aplicación del método de etiquetado de componentes conexas según se ha descrito previamente.



Figura 7. Identificación de objetos.

A partir de los resultados del etiquetado, mostrado en la figura 7, el siguiente proceso, de naturaleza fundamentalmente estética para una mejor visualización, consiste en la asignación de un color diferente según la etiqueta correspondiente. La figura 8, muestra el resultado de este proceso.

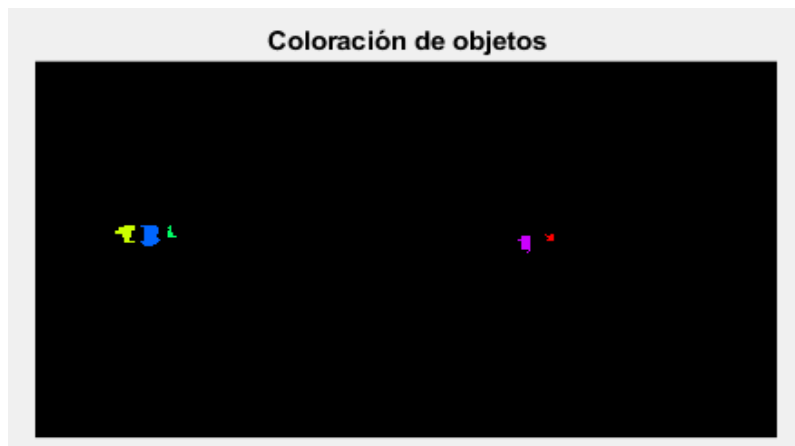


Figura 8. Etiquetado o coloración de objetos.

Finalmente, tras el proceso de etiquetado y coloreado se determina el número diferente de objetos existentes en la imagen, cuyo resultado final se proporciona a través de la GUI de la aplicación, que da soporte a la misma.

3.- Diseño e implementación

3.1 Diseño

Como bien es sabido, los patrones de diseño constituyen un elemento válido y eficiente para desarrollar una interfaz gráfica de usuario intuitiva y de fácil usabilidad, incluyendo la posibilidad de despliegue de herramientas adicionales para su control y manejo (UIPatterns, 2018).

Siguiendo las directrices expresadas previamente, se ha utilizado la herramienta GUIDE de Matlab (2018) para el diseño e integración de la funcionalidad que incluye la aplicación a nivel de interfaz gráfica. Esto permite la fácil edición de las propiedades de los elementos y la posibilidad de autogenerar los contenedores de código correspondientes, como se puede ver en la figura 9, donde se muestra la pantalla correspondiente de dicha herramienta como base del desarrollo.

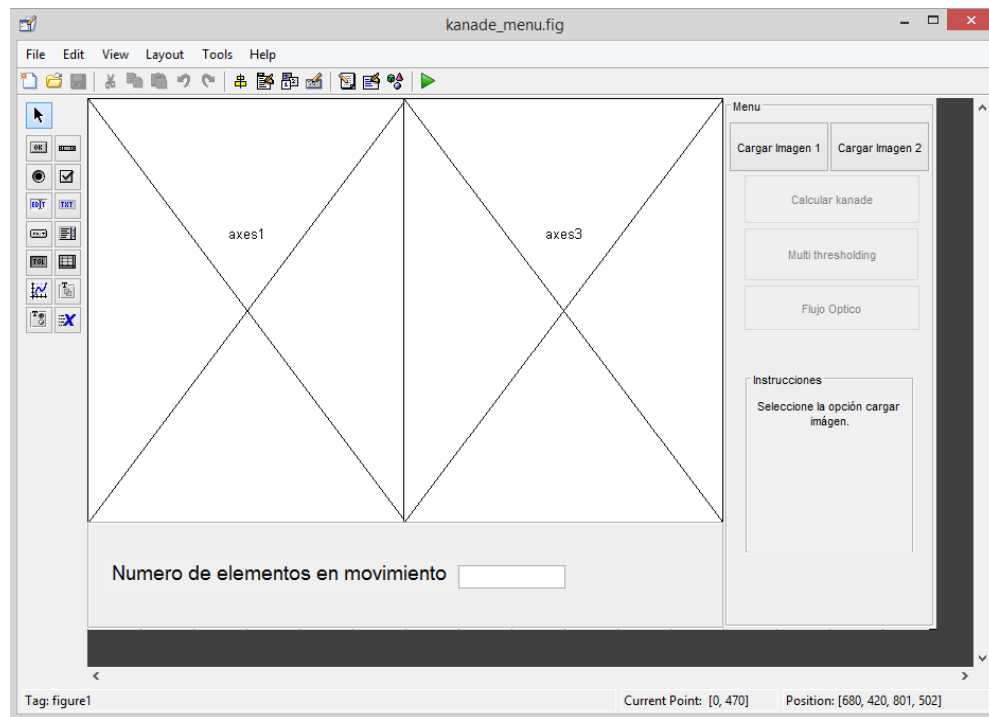


Figura 9 GUIDE de Matlab

Además, como consecuencia del fácil manejo y diseño de este tipo de interfaz, la estructura de un diagrama de clase resulta muy sencilla, ya que permite estructurar la funcionalidad de la aplicación de forma intuitiva, tal y como se muestra en la figura 10, además de posibilitar el añadido de funcionalidades sin cambiar la estructura del diseño.

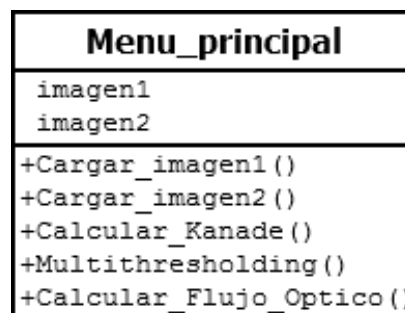


Figura 10 Diagrama de clase

3.2 Implementación

La aplicación se ha desarrollado en Matlab (2018), utilizando el toolbox Image Processing con ayuda de los desarrollos disponibles en Matlab Central (2018). En el Anexo I se proporcionan los detalles más relevantes sobre la implementación del algoritmo de Lucas-Kanade, que determina el movimiento de los objetos en la escena, su ubicación en la imagen, así como el flujo óptico que determina tanto la magnitud del movimiento como el sentido del mismo a nivel de píxel. La base de esta implementación estriba en los conceptos teóricos descritos previamente en la sección 2.2 donde las variaciones espaciales se materializan en forma de primeras derivadas en sendas direcciones x e y mientras que la variación temporal se expresa también en forma de primera derivada entre las dos imágenes bajo análisis. Bajo estas consideraciones y tal como se indica en el mencionado Anexo, estas derivadas se fundamentan en el cálculo de las correspondientes diferencias utilizando una región de vecindad alrededor del píxel bajo análisis, lo que se realiza mediante las correspondientes operaciones de convolución bidimensional, dada la naturaleza también bidimensional de la imagen.

Respecto del proceso de binarización, el propio Matlab (2018) proporciona la función de binarización *multithresh* que implementa el método de Otsu (1979), descrito previamente en la sección 2, si bien en el caso de la implementación realizada en este trabajo para varios umbrales, que es la finalmente utilizada en el presente trabajo. Por otra parte, en relación al proceso de etiquetado de componentes conexas tras la binarización, la función utilizada es la contenida en el toolbox Image Processing de Matlab (2018) como *bwlabel* basada en el método de Haralick y Shapiro (1992). En el caso de la implementación utilizada, para el etiquetado se ha tenido en cuenta una vecindad ocho, que consiste en analizar las conexiones entre píxeles considerando los ocho vecinos en las direcciones horizontal, vertical y diagonal al píxel dado.

Previamente, para la eliminación de ruido el procesamiento se basa en la aplicación de operaciones morfológicas (Pajares y Cruz, 2007), concretamente apertura binaria mediante la función *bwmorph* del toolbox Image Processing de Matlab (2018). Esta operación utiliza un núcleo de vecindad 3x3 para realizar las correspondientes operaciones entre el píxel bajo consideración y los vecinos involucrados.

En la aplicación desarrollada, se permite la carga de imágenes en diferentes formatos de imagen, incluido el conocido como portable network graphics (con extensión png) mediante la correspondiente interfaz gráfica amigable al usuario.

4.- Resultados

En esta sección se consideran y analizan brevemente los resultados obtenidos a partir de la aplicación para diferentes instantes de tiempo en la secuencia de vídeo original. Se consideran cuatro casos que muestran y representan situaciones significativas con vistas al análisis sobre el comportamiento de la aplicación. Estos casos son ilustrativos y reflejan aspectos relevantes ante cada una de dichas situaciones.

a) Caso 1

Caso 1	
Par de imágenes en consideración	130 y 138
Resolución	24 bits
Dimensiones	372x522
Formato	png

En este primer caso, cuya representación se sintetiza en sendas imágenes (a) y (b) de la figura 11, correspondientes a la misma secuencia de vídeo, se pueden observar varios grupos de personas caminando sobre un fondo de edificios. Como siempre, los objetos, personas y áreas con mayor temperatura se muestran con niveles de intensidad mayores respecto del resto de objetos en la escena. Tal es el caso de las personas, que constituyen básicamente los objetos en movimiento bajo análisis.

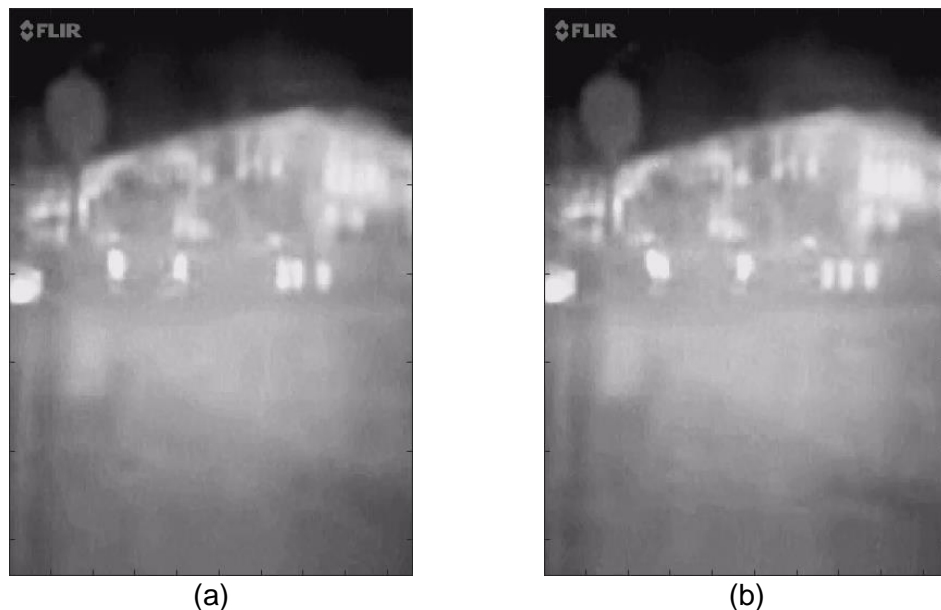


Figura 11 Caso 1: imágenes de una secuencia de vídeo (personas en movimiento).

Tras la aplicación del método de Lucas-Kanade para detectar las diferencias existentes entre ambas imágenes bajo análisis. El resultado es el mostrado en la figura 12, donde se puede observar los elementos destacados en movimiento de la escena.



Figura 12 Diferencias entre dos imágenes mediante Lucas-Kanade.

Una vez identificados los objetos y las zonas de interés correspondientes asociadas al movimiento, se aplica el procedimiento de binarización mediante detección de umbral, para obtener la imagen mostrada en la figura 13(a). Sobre esta imagen se aplican las técnicas previstas para eliminación de ruido y píxeles espurios obteniendo el resultado mostrado en la figura 13(b). A partir de la cual se obtienen los objetos etiquetados mediante el método basado en componentes conexas, obteniéndose el resultado mostrado en la figura 13(c) en la que estos objetos han sido coloreados en función de sus diferentes etiquetas asignadas.

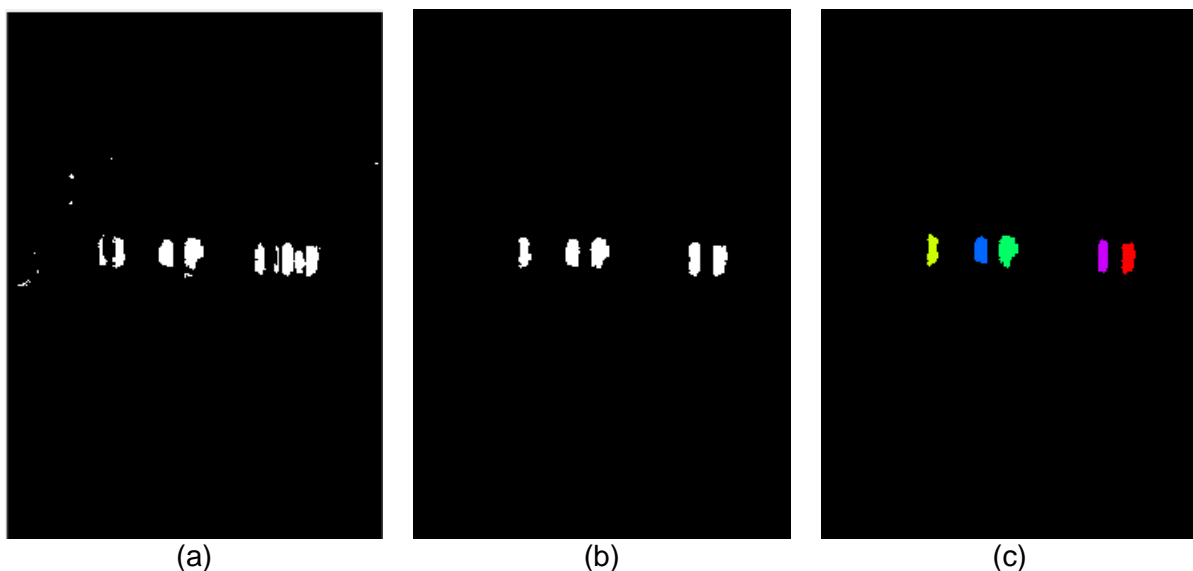


Figura 13 (a) Imagen binarizada a partir de la imagen de la figura 12; (b) la misma imagen sin ruido; (c) imagen con los objetos etiquetados e identificados por color.

A partir de los resultados mostrados, los objetos en movimiento han sido detectados, si bien, dada la baja resolución de este tipo de imágenes, no es posible una clara identificación de los objetos como tal, más allá de la existencia de su movimiento. En la figura 14 se obtiene finalmente la representación del flujo óptico, observándose el sentido y dirección de éste según la representación gráfica de los correspondientes vectores en cada píxel. Obsérvese el flujo en la zona de los objetos, figura 14, así como en la parte inferior de la imagen, en este caso debido al movimiento de la cámara y la mayor proximidad de esta zona a la cámara.

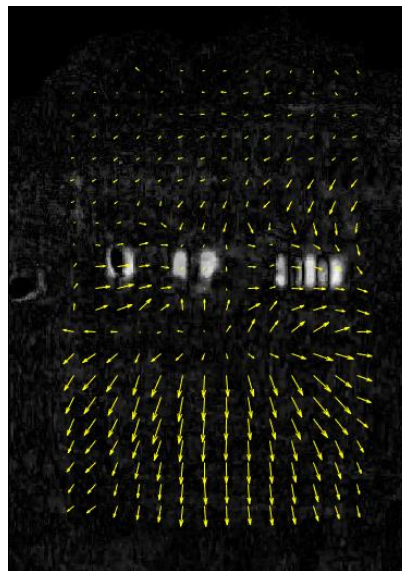


Figura 14 Flujo óptico para las imágenes del caso 1.

b) Caso 2

Caso 2	
Par de imágenes en consideración	155 y 157
Resolución	24 bits
Dimensiones	372x522
Formato	png

En este segundo caso se tienen en consideración las imágenes número 155 y 157 del vídeo, observándose ahora la presencia de un vehículo que se cruza con un grupo de personas, en este caso también en movimiento. La situación que se pone de manifiesto es la diferente velocidad del vehículo frente al grupo de personas caminando, lo que está relacionado con la derivada temporal. Además, el diferente volumen de estos elementos en

la escena, que tiene que ver con las variaciones espaciales y por tanto con la aplicación de las derivadas de esta naturaleza.

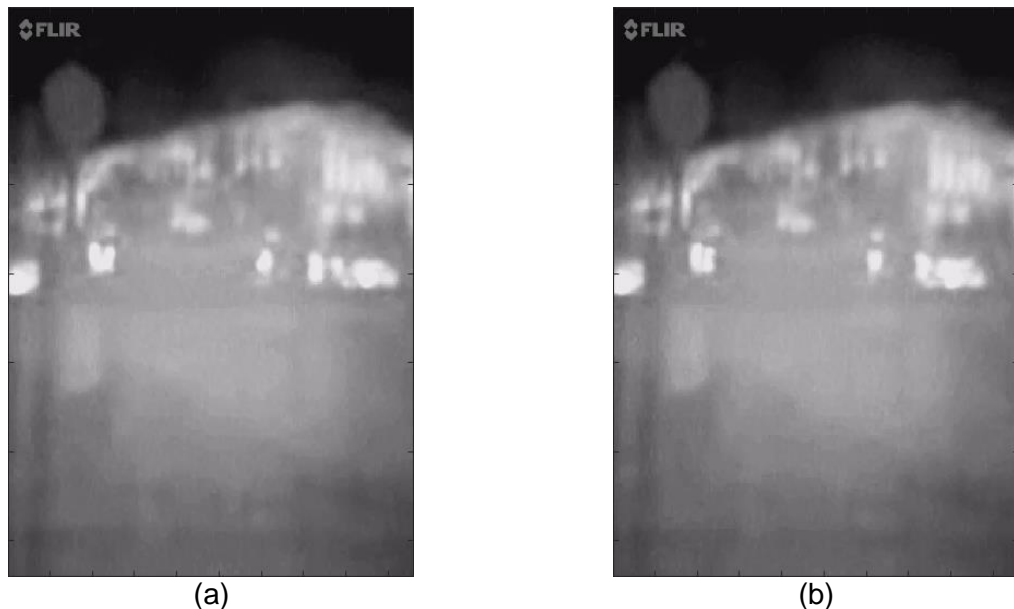


Figura 15 Caso 2: imágenes de una secuencia de vídeo (personas y vehículo en movimiento).

Tras aplicar el método de Lucas-Kanade se obtiene la imagen de la figura 16.



Figura 16 Diferencias entre dos imágenes mediante Lucas-Kanade.

Una vez identificados los objetos y las zonas de interés correspondientes asociadas al movimiento, se aplica como en el caso anterior el procedimiento de binarización mediante detección de umbral, figura 17(a). A partir de este resultado se obtienen sendas imágenes sin ruido, figura 17(b) y con las correspondientes etiquetas, figura 17(c). Obsérvese en este

caso la agresividad del proceso de eliminación de ruido que ha suprimido el movimiento en la parte izquierda de la imagen. En la parte derecha de la imagen mostrada en la figura 17(c) aparece el vehículo claramente identificado frente al grupo de personas situadas en la misma región de la escena.

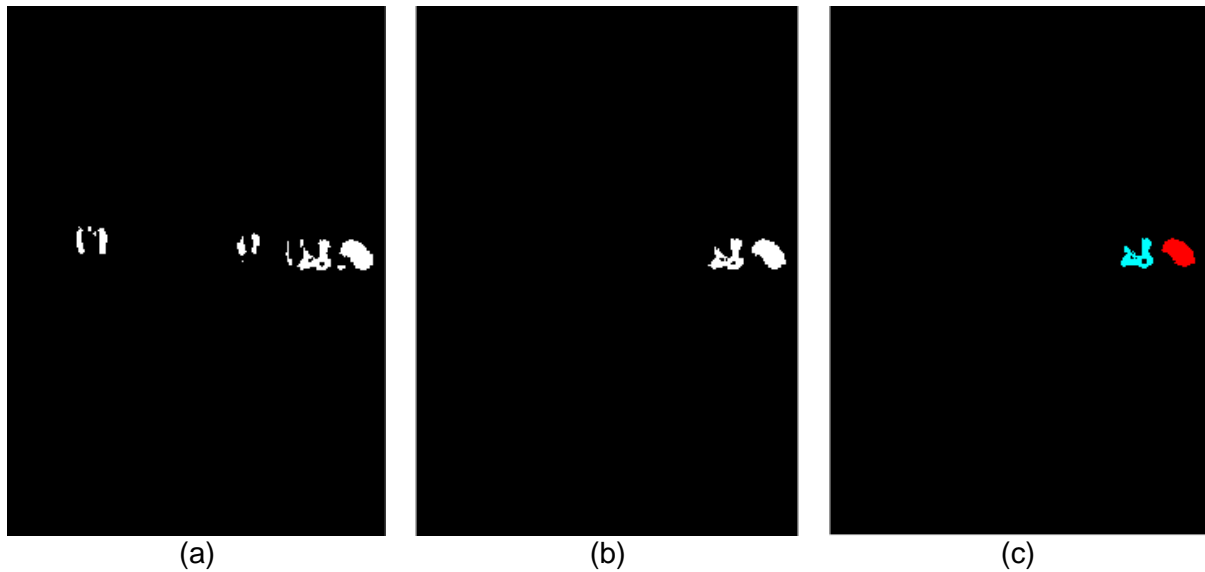


Figura 17 (a) Imagen binarizada a partir de la imagen de la figura 16; (b) la misma imagen sin ruido; (c) imagen con los objetos etiquetados e identificados por color.

Como en el caso anterior, el cálculo del flujo óptico se ve afectado por el ruido de la imagen en la parte inferior, observándose en cualquier caso la presencia del vehículo que se desplaza en dirección contraria al grupo de personas.

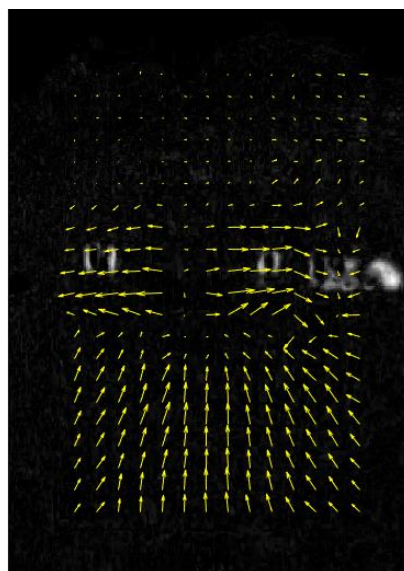


Figura 18 Flujo óptico para las imágenes del caso 2.

c) Caso 3

Caso 3	
Par de imágenes en consideración	168 y 171
Resolución	24 bits
Dimensiones	372x522
Formato	png

En este tercer caso se tienen en consideración el par de imágenes dado por los *frames* 168 y 171, en los que se observa el grupo de personas del caso anterior y el vehículo mejor diferenciado, una vez éste ha superado el grupo de personas a la derecha. Se observa también un movimiento de personas en la parte izquierda de las imágenes. Como en el caso anterior, se trata de movimiento de objetos diversos, con diferentes tamaños espaciales y diferentes velocidades relacionadas respectivamente con las variaciones espaciales y temporales, que captan los movimientos según las mencionadas direcciones.

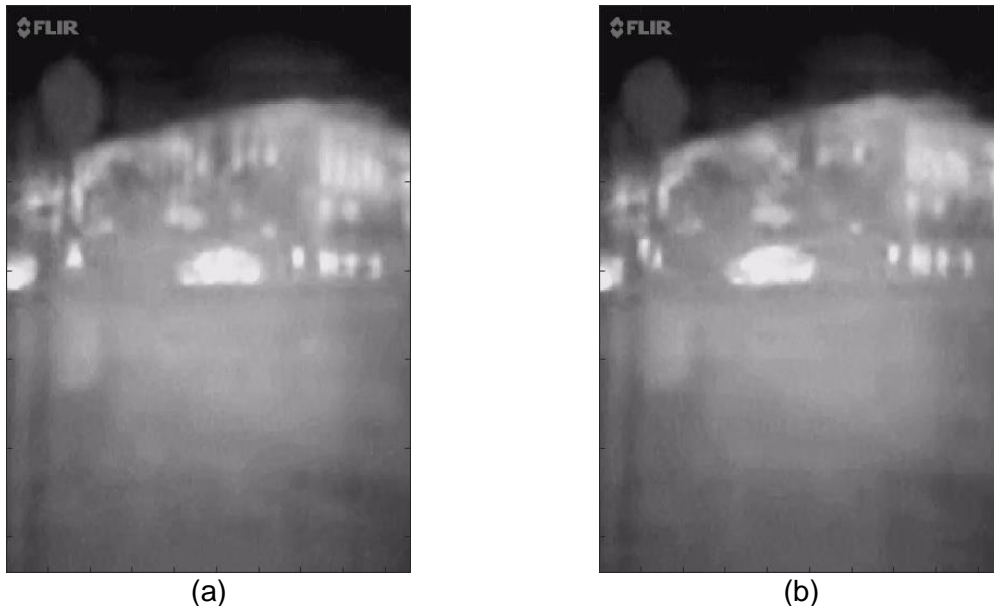


Figura 19 Caso 3: imágenes de una secuencia de vídeo (personas y vehículo en movimiento con el grupo de personas superado).

Como en los casos anteriores, tras aplicar el método Lucas-Kanade se obtiene la imagen de la figura 20. En ella se observa claramente el vehículo frente al grupo de personas en la parte derecha de las imágenes que, en este caso, aparece más difuminado debido a que el movimiento de las personas en este caso es menos acusado que en los casos anteriores, obviamente debido a una disminución en la velocidad de movimiento. En la parte izquierda aparece un nuevo grupo de personas, que se identifican con claridad. Por el contrario, en el fondo de la escena aparecen diferencias significativas, debido

naturalmente al movimiento de la cámara durante el momento de captura de la secuencia del vídeo. Hay que tener en cuenta en relación a este aspecto que las imágenes se capturaron con la cámara en movimiento con un desplazamiento de ésta avanzando y aproximándose hacia los mencionados objetos en la escena.



Figura 20 Diferencias entre dos imágenes mediante Lucas-Kanade.

Aplicando los mismos procedimientos que en los dos casos anteriores, se obtienen las imágenes mostradas en las figuras 21 (a) y (b). En el primer caso se muestra la imagen binarizada con el ruido eliminado. Obsérvese en ella la gran cantidad de regiones identificadas como portadoras de movimiento debido al mencionado movimiento de la cámara. Como consecuencia de ello, el número de objetos y regiones etiquetadas aumenta considerablemente en relación a los casos anteriores.



(a)



(b)

Figura 21 (a) Imagen binarizada a partir de la imagen de la figura 20; (b) imagen con los objetos etiquetados e identificados por color.

El cómputo del flujo óptico a partir de las imágenes anteriores se muestra en la figura 22, observándose el efecto de los movimientos de la cámara señalados e identificando claramente la presencia del vehículo por la magnitud del movimiento asociado.

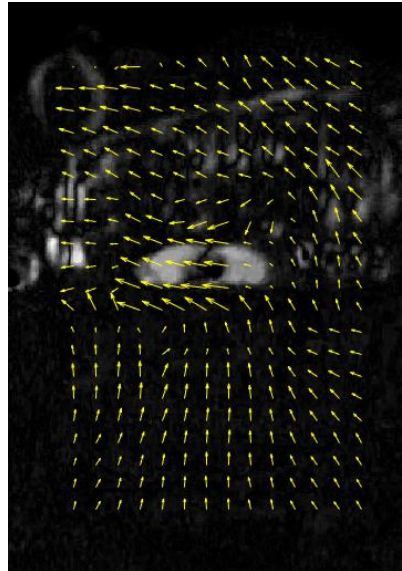


Figura 22 Flujo óptico para las imágenes del caso 3.

d) Caso 4

Caso 4	
Par de imágenes en consideración	42 y 43
Resolución	24 bits
Dimensiones	372x522
Formato	png

En el cuarto caso se tienen en consideración las imágenes que se corresponden con la número 42 y 43 de la misma secuencia de vídeo anterior. En este caso, y observando a simple vista el par de imágenes analizados, sobre la escena aparecen dos objetos voluminosos sin movimiento. Se trata de dos vehículos, un utilitario y un autobús estacionados frente a una construcción donde también se aprecian elementos diferenciados por su emisividad térmica. Teóricamente en este caso se trata de considerar una escena estática donde no existe movimiento aparente de los objetos. El único movimiento es el derivado del movimiento de la cámara durante la captura de las imágenes y por tanto se trata de un movimiento subyacente, que constituye un aspecto diferente para la consideración de este tipo de escenas en determinadas aplicaciones tales como video-vigilancia, donde hay que tener en cuenta estos hechos para evitar el lanzamiento de falsas

alarmas debidas a los movimientos espurios frente a movimientos de cierta intensidad, que si constituirían señales de alarma reales por situaciones tales como intrusiones de elementos en la escena bajo vigilancia.

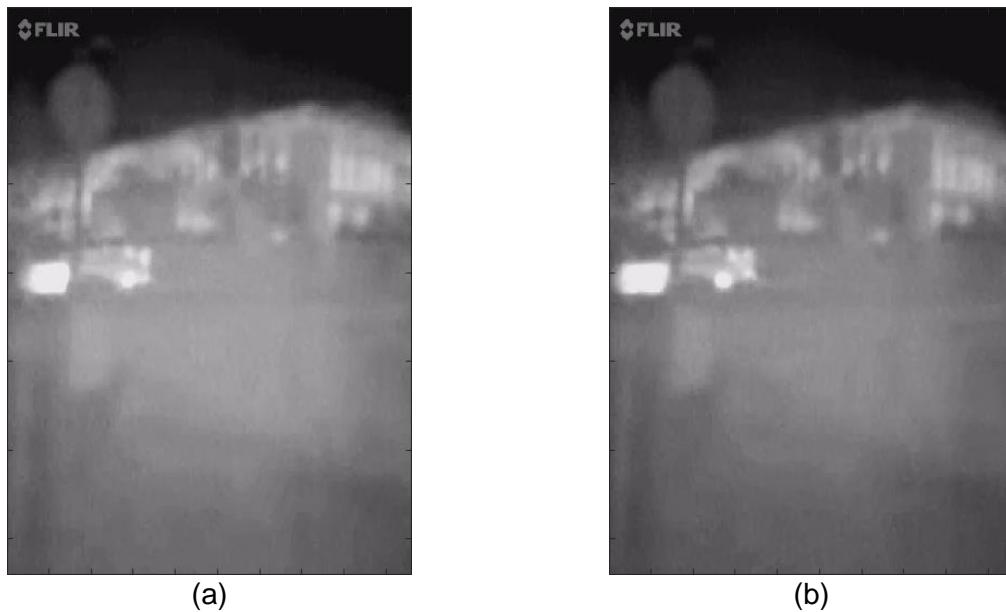


Figura 23 Caso 4: escena con todos los elementos estáticos.

La aplicación del método de Lucas-Kanade genera el resultado mostrado en la imagen de la figura 24. En este estado del proceso se observa en este caso la presencia de abundantes diferencias puntuales, derivadas exclusivamente del movimiento de la cámara.



Figura 24 Lucas-Kanade Caso 4

La aplicación de los procesos de binarización y eliminación del ruido da como resultado las imágenes mostradas en las figuras 25(a) y (b) respectivamente, observándose

en esta caso cómo prácticamente toda la escena aparece como con movimiento, ocasionando como consecuencia la ineficiencia desde el punto de vista del etiquetado de componentes conexas, dada la gran extensión de la región inferior de la imagen que no se soluciona con la eliminación del ruido.

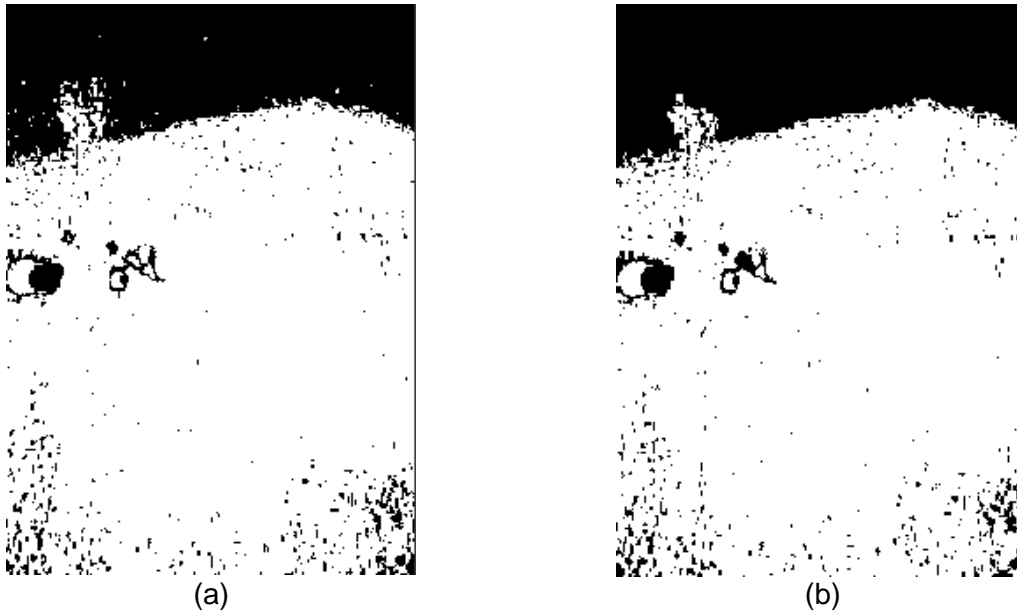


Figura 25 (a) Imagen binarizada a partir de la imagen de la figura 24; (b) imagen binaria con eliminación de ruido.

Como consecuencia de lo anterior, y aún tras la eliminación del ruido, el cómputo del flujo óptico genera la imagen mostrada en la figura 26. Obsérvese en este caso, cómo la única presencia significativa relativa a la magnitud del flujo óptico aparece en la parte inferior de la imagen, naturalmente como consecuencia del movimiento de la cámara. Por el contrario, y a pesar de la falta de identificación de objetos en la escena, como se ha apuntado previamente, gracias al flujo óptico se determina la falta de movimiento significativo, algo digno de mención desde el punto de vista de detección del movimiento, no así desde la identificación de objetos como se acaba de exponer.

En efecto, desde esta perspectiva, y por ejemplo en el caso de una aplicación basada en video-vigilancia, es importante llegar a esta conclusión con el fin de descartar movimientos significativos en la escena, lo que indicaría una situación estática que no conllevaría en ningún caso la activación de ningún dispositivo o alarma previstos para garantizar la seguridad en estos casos.

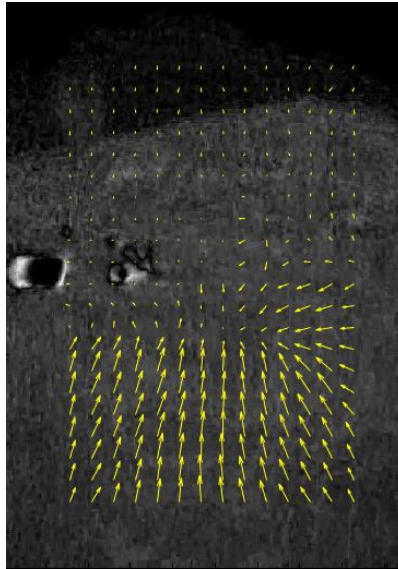


Figura 26 Flujo óptico para las imágenes del caso 4.

5.- Conclusiones y trabajo futuro

En el presente trabajo se exponen los detalles relativos al desarrollo de la aplicación que se plantea sobre la detección de movimiento en imágenes térmicas con identificación de los objetos que originan el movimiento.

Con tal propósito se han implementado dos métodos clave relativos por un lado a la detección del movimiento y consecuentemente el flujo óptico a través del algoritmo de Lucas-Kanade y por otro lado a la detección de los objetos causantes del movimiento mediante técnicas de umbralización (multi-umbral) y etiquetado de componentes conexas.

La aplicación se ha desarrollado en lenguaje Matlab bajo las pautas establecidas por la Ingeniería del Software, bajo un GUI amigable que permite al usuario su utilización de forma eficiente.

La utilización de técnicas orientadas a la detección o no de movimiento a través del flujo óptico han resultado eficientes, incluso ante la ausencia de movimiento como ocurre en el caso 4 analizado en la sección cuatro. No así la identificación de objetos que en este caso resulta inviable. Esto nos lleva a pensar que cuando las diferencias de intensidad entre dos imágenes de la secuencia es alta en cuanto al número de píxeles, aunque no lo sea en intensidad, la identificación de objetos no resulta tan eficiente como en los casos en que esto no ocurre, como los analizados en los casos 1 a 3 en la misma sección cuatro.

Si bien se han alcanzado los objetivos propuestos para el desarrollo del presente trabajo, se han identificado diversos problemas a la hora de tratar con imágenes de tipo térmico.

Dado que la temperatura afecta a la intensidad de la imagen y como se ha visto en la memoria ésta es una característica fundamental a la hora de la detección de movimiento, un objeto que no emite calor podría estar moviéndose sin ser detectado. Por este mismo motivo se propone como líneas futuras el hecho de combinar la detección de movimiento tanto en imágenes térmicas como en imágenes procedentes de otro rango espectral, principalmente color en situaciones donde el color sea relevante, por ejemplo en condiciones de iluminación efectivas, no así por ejemplo durante la noche en las que las cámaras operando en el espectro visible no resultan operativas.

Otro de los factores problemáticos es la habitual baja resolución de las imágenes térmicas, lo que produce que objetos lejanos sean difíciles de reconocer y/o detectar ya que se pueden confundir con ruido de la imagen.

Pese a estos problemas se ha conseguido implementar una aplicación intuitiva, mantenible y fácilmente extensible que sienta la base para una futura automatización del proceso o su extrapolación a otros sistemas. Por esta razón la aplicación se ha diseñado de forma flexible y modular con el fin de poder incorporar nuevas funcionalidades sin un esfuerzo significativo.

Otro aspecto a considerar es la automatización del proceso, de forma que el procesamiento se realice directamente a partir del vídeo original proporcionado por una cámara que observa la escena. De esta forma, es posible su aplicación efectiva en operaciones de vídeo-vigilancia por ejemplo sin intervención humana, de forma que en función del movimiento detectado y la naturaleza de éste se levante la correspondiente alarma de detección de movimiento y ubicación.

6.- *Conclusions and future work*

In the present work we expose details related to the development of the application that arises on the detection of movement in thermal images with identification of objects causing the movement.

For this purpose, two key methods have been implemented, one relating to the detection of movement and consequently the optical flow through the Lucas-Kanade algorithm and, on the other hand, the detection of movement caused by objects through thresholding techniques (multi -threshold) and labelling of connected components.

The application has been developed in Matlab language under the guidelines established by the Software Engineering, under a friendly GUI that allows the user to use it efficiently.

The optical flow detection is a useful approach with efficient performance even if there is no movement in the scene. This has been shown in case 4 (section four), where the optical flow indicates no movement in the scene. Nevertheless, in this case object identification is inefficient due to the movement produced by the camera movement, that generates local movements at pixel level and the objet identification and labelling is not possible. This does not occur in the other cases (1 to 3) analyzed, where the performance of the proposed approach becomes sufficient for the intended purpose.

Although the objectives proposed for the development of this work have been achieved, several problems have been identified when dealing with thermal images.

Since the temperature affects the intensity of the image and as seen in memory the intensity level is a main feature for motion detection, an object that does not emit heat could be moving without being detected. For this reason, as future lines of work could be the combination of both, thermal images and images coming from another spectral range, mainly colour in situations where the colour is relevant, for example in effective lighting conditions, not for example during the night in which the cameras operating at the visible spectrum are not operative.

Another problematic factor is the usual low resolution of thermal images, which causes distant objects to be difficult to recognize and detect since they can be confused with image noise.

Despite these problems, it has been possible to implement an intuitive, maintainable and easily extensible application that forms the basis for future automation of the process or its extrapolation to other systems. For this reason, the application has been designed in a flexible and modular way in order to be able to incorporate new features and functionalities without significant effort.

Another aspect to consider is the automation of the process, so that the processing is done directly from the original video provided by a camera that observes the scene. In this way, its effective application in video surveillance operations is possible, for example, without human intervention, so that depending on the movement detected and thereof the nature, the corresponding detection movement alarm is triggered.

Bibliografía

- 1 FLIR ONE © (2018). Disponible on-line: <http://www.flir.com/flirone/> (accedido Mayo 2018).
- 2 Haralick, R.M., Shapiro, L.G. (1992). Computer and Robot Vision. Volume I, Addison-Wesley, pp. 28-48.
- 3 Lucas B.D., Kanade T. (1981). *An iterative image registration technique with an application to stereo vision*. En *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI '81)*, Abril, 1981, pp. 674-679.
- 4 MatlabCentral. *File Exchange*. Disponible on-line: <https://es.mathworks.com/matlabcentral/fileexchange/> (accedido Mayo 2018).
- 5 Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. IEEE Trans. Sys., Man, and Cybernetics, 9(1), 62-66.
- 6 Pajares G., Cruz J.M., (2007). Visión por computador: Imágenes digitales y aplicaciones. Editorial RA-MA. Segunda edición. Madrid
- 7 The Mathworks. Matlab. Disponible on-line: <https://es.mathworks.com/> (accedido Mayo 2018).
- 8 UIPatterns (2018). Design patterns. Disponible on-line <http://ui-patterns.com/> (accedido Mayo 2018).

Anexo I: Código base Lucas-Kanade

A nivel de la aplicación y del código para la implementación del cálculo del flujo óptico se siguen los siguientes pasos, cuyo desarrollo está fundamentado en los métodos incluidos en Matlab Central (2018).

Para cada punto se calcula $I(x)$, $I(y)$ e $I(t)$ y sus correspondientes derivadas parciales:

```
% Cálculo de las derivadas fx, fy y ft
ho = [-1 0 1; -2 0 2; -1 0 1]; hv = [-1 -2 -1; 0 0 0; 1 2 1];
hu = ones(3);

fx = conv2(im1g, ho, 'same') + conv2(im2g, ho, 'same');
fy = conv2(im1g, hv, 'same') + conv2(im2g, hv, 'same');
ft = conv2(im1g, hu, 'same') - conv2(im2g, hu, 'same');
```

Se obtienen los valores para A , b y v :

```
u = zeros(M,N);
v = u;
for i = 2:1:M-1
    for j = 2:1:N-1

        fx1 = fx(i-1:i+1, j-1:j+1);
        fy1 = fy(i-1:i+1, j-1:j+1);
        ft1 = ft(i-1:i+1, j-1:j+1);

        a11 = sum(sum(fx1.*fx1));
        a12 = sum(sum(fx1.*fy1));
        a21 = a12;
        a22 = sum(sum(fy1.*fy1));
        b1 = sum(sum(ft1.*fx1));
        b2 = sum(sum(ft1.*fy1));

        A = [a11 a12; a21 a22];
        B = [b1; b2];

        U = pinv(A'*A)*A'*B;

        u(i,j)=U(1);
        v(i,j)=U(2);
    end
end
```

Se aplican dos descomposiciones piramidales gaussianas para reducir la dimensión de las matrices u y v además de obtener las coordenadas de u y v en la imagen original para así poder dibujar y visualizar los vectores de velocidad.

```
% Visualización del flujo óptico

% 1) Cambiamos las filas de orden
u = flipud(u); v = flipud(v);

%2) Aplicamos el filtro de la mediana en vecindades [5,5]
mu = medfilt2(u,[5 5]); mv = medfilt2(v,[5 5]);
```

```
%3) Aplicamos dos descomposiciones piramidales gaussianas para reducir la
%dimensión de las matrices u y v
ru = reducir(reducir(mu)); rv = reducir(reducir(mv));

escala = 0; %valor por defecto (escalado de las flechas de vectores)
figure; quiver(ru, -rv, escala, 'r', 'LineWidth', 2), %ejes iguales
```

En la figura A1.1 se muestra un ejemplo de visualización del flujo óptico, resultado de los procesos indicados.

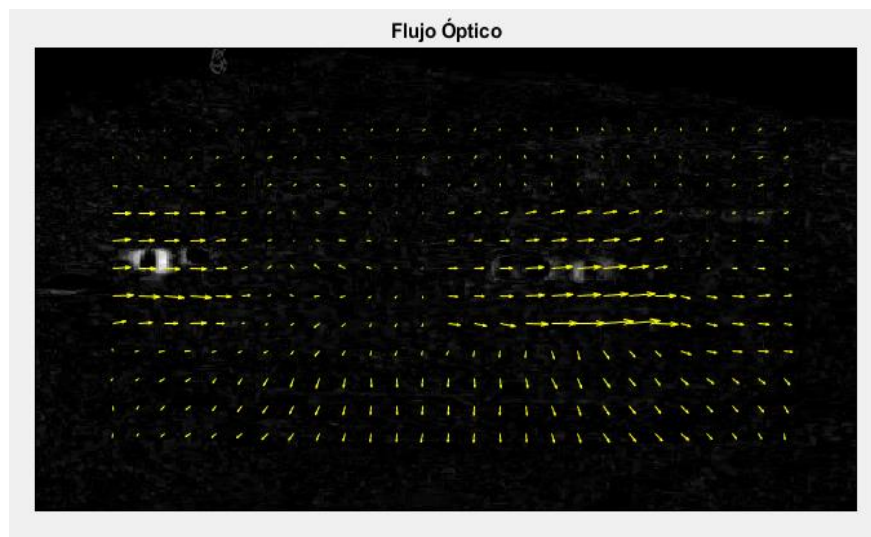


Figura A1.1 Visualización del Flujo Óptico.

Anexo II: Manual de usuario

El presente apéndice tiene como objetivo describir el manejo de la aplicación mediante el correspondiente Manual de Usuario, que proporciona los detalles de uso de la misma.

La figura A2.1 muestra el panel de inicio de la aplicación, como se puede observar solamente se puede proceder a la carga de las imágenes, y una vez realizada ésta operación se habilitan el resto de los controles del usuario

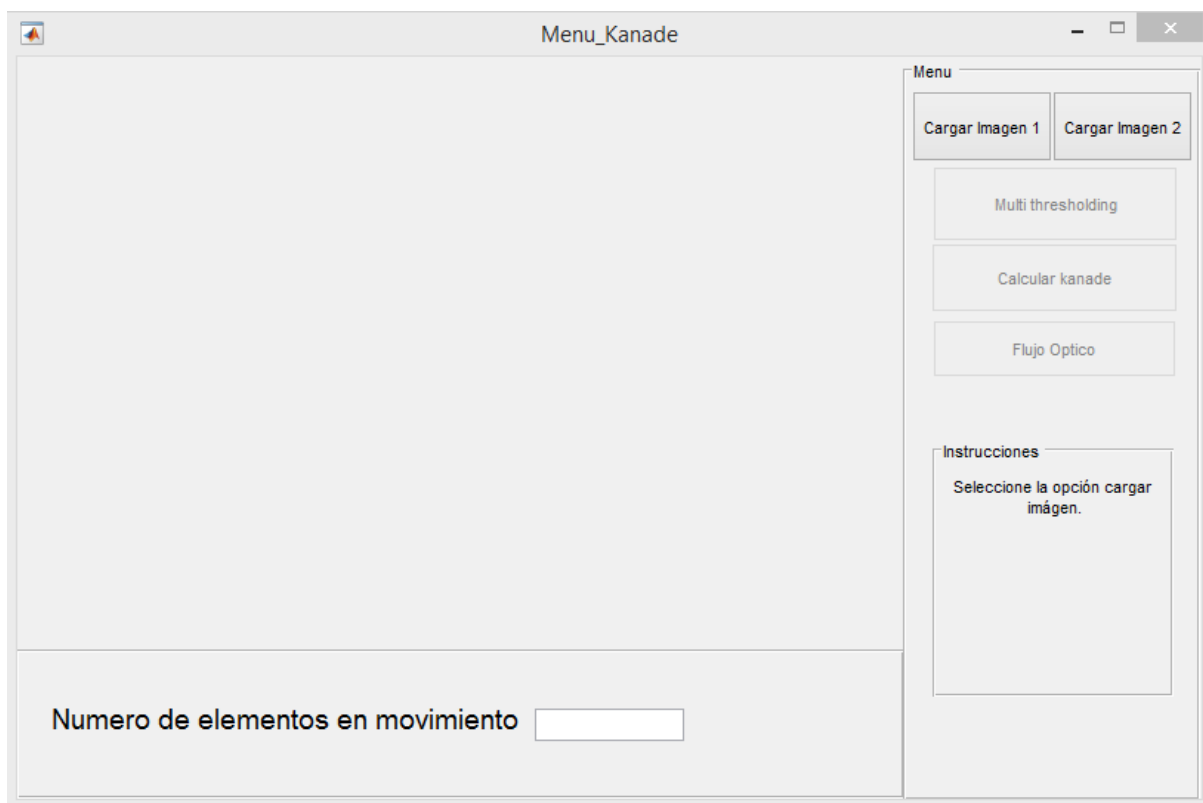


Figura A2.1 Pantalla inicial de la aplicación.

El primer paso consiste en la carga de las imágenes (*frames*) para su uso en la aplicación, para ello se utilizan dos botones “Cargar Imagen 1” y “Cargar Imagen 2”.

El proceso de carga se muestra en la figura A2.2, junto con la visualización progresiva de las imágenes cargadas, tal y como se muestra en las figuras A2.3 y A2.4 donde aparecen las distintas imágenes de la secuencia

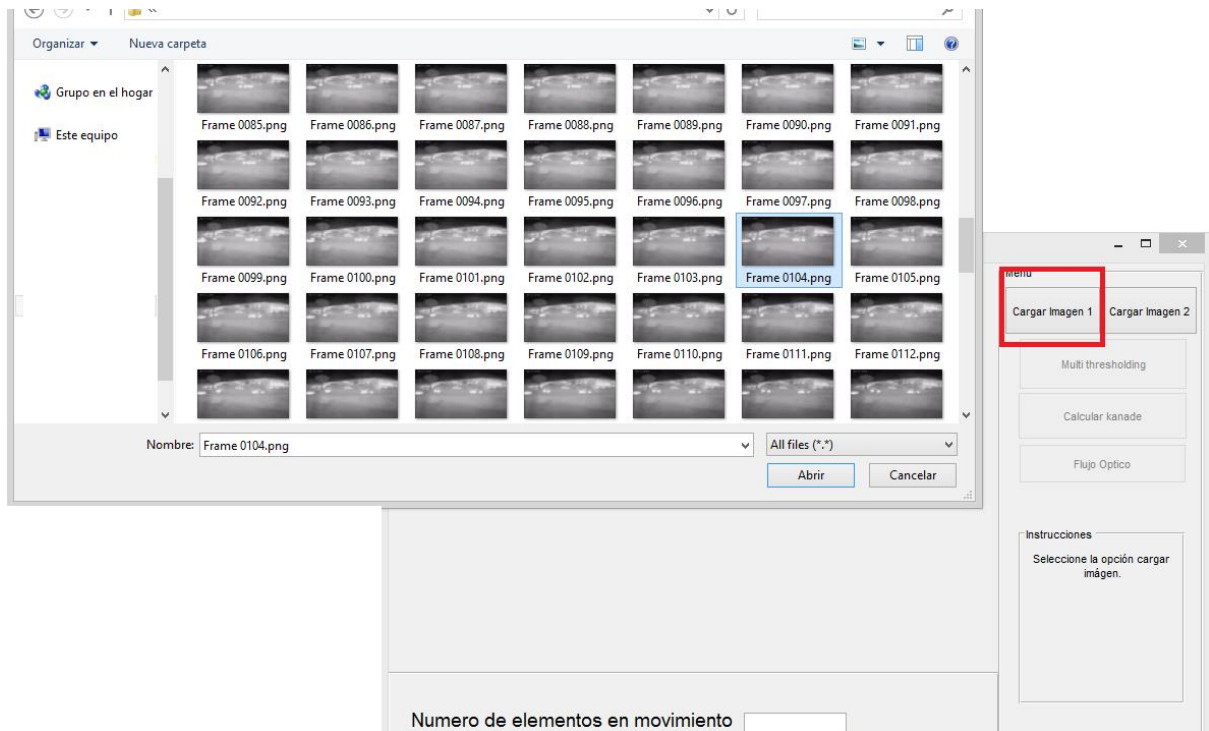


Figura A2.2 Carga de imágenes.



Figura A2.3 Visualización de la imagen de referencia cargada.



Figura A2.4 Visualización de las dos imágenes cargadas.

Tras la carga de las imágenes se activa el botón de la GUI que permite el tratamiento de las imágenes con el algoritmo de Lucas-Kanade. Una vez se pulsa el botón “Calcular Kanade” se muestra el resultado que aparece en la figura A2.5.



Figura A2.5 Imágenes base para aplicación del algoritmo de Lucas-Kanade.

Una vez ejecutado el método de Lucas-Kanade se activa tanto el botón “MultiThresholding”, que permite realizar los tratamientos pertinentes de la imagen explicado con anterioridad en la memoria, como el botón “Flujo Óptico”, que calcula el flujo óptico en la imagen y lo muestra sobre el GUI, tal como se puede ver en la figura A2.10, tras los correspondientes procesos de binarización (figura A2.6), eliminación de ruido (figura A2.7), identificación de objetos (figura A2.8) y etiquetado o coloreado de los objetos detectados (figura A2.9).

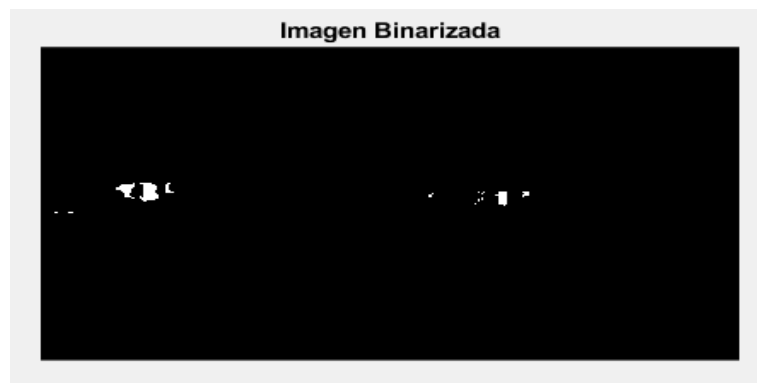


Figura A2.6 Imagen binarizada.

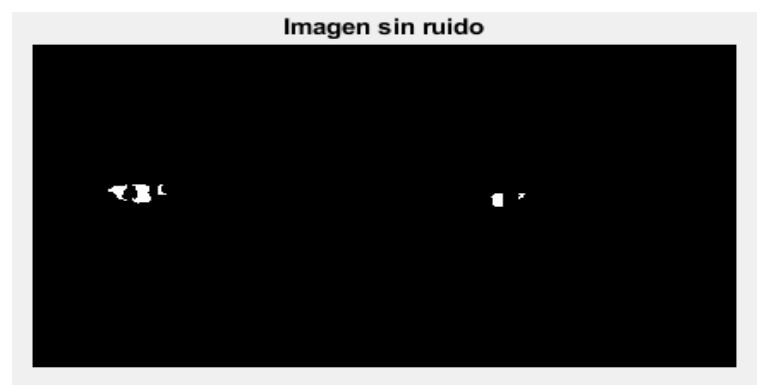


Figura A2.7 Imagen filtrada para eliminación de ruido.



Figura A2.8 Identificación de objetos.

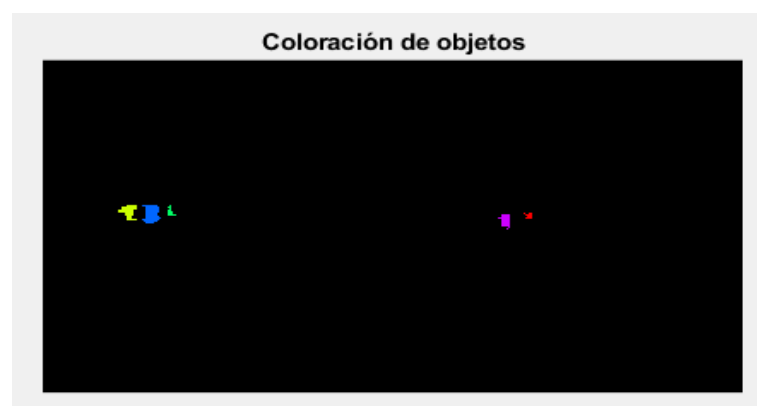


Figura A2.9 Etiquetado de objetos identificados.

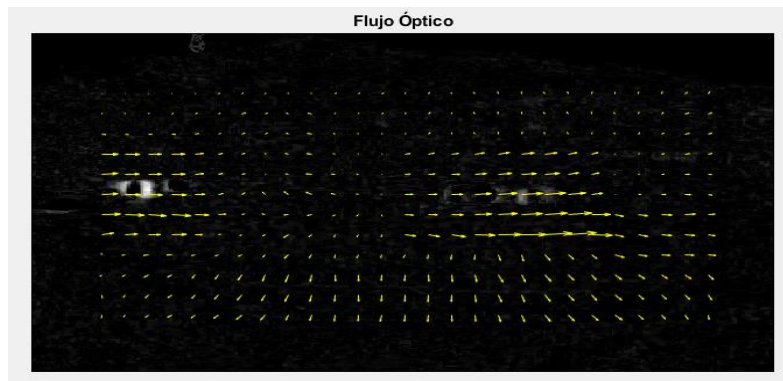


Figura A2.10 Visualización del flujo óptico.

Tras la ejecución del botón “Multi thresholding” se muestra en la parte inferior de la GUI el conteo de los objetos en movimiento que se han detectado en las Imágenes, como se puede observar en la figura A2.11.



Figura A2.11 Número de objetos en movimiento identificados.